Response to Washington State RFI

Modernized Election Systems/OSOS RFI 16-04

December 23, 2015

Dr. Joe Kiniry, Galois, Inc. kiniry@galois.com
Dr. Dan Zimmerman, Galois, Inc. dmz@galois.com

Responder:

Galois, Inc.
421 SW Sixth Avenue
Suite 300
Portland, OR 97204

Contact:

Jodee LeRoux
ph 503.808.7209
fax 503.350.0833
jodee@galois.com

Galois and its elections-centric spin-out do not anticipate making an independent response to the eventual RFP, because our focus is on high-assurance core election systems rather than their corresponding management systems—however, we are open to partnering with other vendors. We are responding to this RFI because we have a broad interest in helping to improve the quality and decrease the cost of elections nationwide. Toward that end, we have a number of suggestions to help in the creation of a stronger RFP or set of RFPs that we believe will result in a significantly better end result for Washington State and its citizens.

If the RFP were more modular than this RFI (as discussed later in this response), we would be interested in responding to components that align with our interests and expertise in high-assurance secure systems. For example, if the proposed monolithic RFP were released as a set of component RFPs, we would be interested in proposing to RFPs focusing on: (a) the design and implementation of secure protocols and subsystems maintaining the aspects of the Washington IT Security Policy relevant to elections; (b) red teams analyzing the quality, security, correctness, or availability practices, policies, processes, methodologies, or results of the systems engineering of other contractors; and (c) QA teams testing the subsystems engineered by other vendors, particularly if that testing focuses on correctness and security issues that necessitate systems being classified as mission-critical.

There are two main points that we believe need to be addressed so that Washington State and its voters get the greatest benefit from the eventual system. These are verifiability and the elimination of vendor lock-in.

Verifiability is the idea that individual voters, including elections officials and candidates, should be able to obtain evidence that their votes were correctly interpreted, counted, and included in the final result of the election. There is a large amount of distrust in current voting systems, which is only compounded by the major public and private sector security breaches we hear about in the media every week, and much of that distrust would be mitigated by verifiability.

Vendor lock-in is a huge problem for jurisdictions that have little recourse when they end up with a system that does not meet their needs. A competitive marketplace favors the consumer, so the RFP should favor proposals that do not eliminate future competition.

With these two goals in mind, there are 6 points that we believe should be prioritized in the RFP:

- preconditions on potential applicants;
- open source software;
- open APIs and data formats;
- commercial off-the-shelf (COTS) hardware;
- evidence of correctness for unsupervised voting; and
- third party audits and verification.

The remainder of this response will focus on each of these points, highlighting how they achieve the goals of *verifiability* and *elimination of vendor lock-in*.

1 Preconditions on potential applicants

The vast majority of RFPs we see in the public elections space have preconditions on potential applicants that essentially eliminate all competition. Example preconditions we commonly see are of these forms: applications must have run elections in our state for X years; applicants must have implemented and deployed elections systems for Y years; applicants must have products already certified at the federal level; etc. Each of these conditions poses a nearly insurmountable barrier to entry for any new vendor, including reputable vendors with years of relevant experience in other countries or domains adjacent to elections. Such RFPs might as well state that the only applicants that need apply are the oligopoly of existing elections vendors in the U.S.A.

We suggest, as an alternative to these typical blanket preconditions, that you stipulate the following requirements for applications: national or international experience in elections; expertise in designing, implementing, and deploying mission-critical systems; peer-reviewed evidence that their work is best-in-class; and strong referrals at the local, state, and federal level relevant to mission-critical systems.

These alternate high standards open up the RFPs to a far larger set of competitors while raising the bar on applicants' capabilities. Both of these results help with procuring a high-quality election system at a reasonable cost.

2 Open source

Open source software helps with verifiability by allowing individuals to inspect and understand the voting system. It also encourages security practices that are worthy of trust. Based upon numerous audits, software leaks, and Freedom of Information Act responses, it is clear that existing, closed source voting systems rely on attackers not knowing details about their implementations. If existing systems were made open source, it is extremely likely that many vulnerabilities would be rapidly discovered. While some of these may be simple programming errors that can be fixed, others are security properties that rely on the source remaining secret. In general this allows for lazy design, which makes it cheaper and easier for vendors to create software quickly at the cost of good security measures. Open source strongly discourages this sort of design.

Software that uses secret source as a security principle has a frightening implication for the customers of the software. They must trust an unknown number of people who participated in the creation of the software to keep the secret and hold it responsibly. Through careful design this fundamental flaw is avoided, particularly with proper application of cryptography.

Open software can be as secure as any closed software. Developers of open source software are not tempted to rely on false security from secret source, instead opting for much stronger guarantees of security, including mathematical proof. There are many responsible researchers, particularly in the area of elections technology, who will inspect open source systems, report bugs to the developer, and often even submit fixes when they find them. These statements are not theoretical. In fact, virtually all the technology that provides some degree of security in all of our computers, smartphones, and online secure transactions is open source.

The nature and delivery of open source software must be mandated in a clear and unambiguous fashion in RFPs. The interpretation of such open source mandates has been abused by duplicitous vendors in numerous past systems developed for public authorities. Companies commonly follow the letter of the contract rather than its true intent.

More precisely, we recommend that open source systems be designed, implemented, validated, verified, integrated, and deployed in a public fashion on a standard open source collaboration platform like GitHub.[1] All artifacts—not just source code—associated with the development of the system must be made public early and often. Vendors love to simply "throw over the wall" enormous undocumented software systems, claiming that they have released open source systems when, in fact, they have abused the nature of open source, obstructed the parties interested in improving the quality of the systems in question, and thumbed their noses at the clients that cared about transparency in the first place. A pile of code with no associated documentation on its requirements, quality assurance, testing, usability, etc. is nearly as bad as a closed source system.

A common objection to open source on the part of software providers is that open source software will be freely copied and used by anyone, not just their customers, causing them to lose revenue. While there are open source models that allow free copying and use, such as the "free software" model used by the Linux operating system and many other open source projects, there are also open source models with more restrictive licensing. Software providers can maintain viable revenue streams with free or restrictive open source licenses; for example, Red Hat[2] is a successful company that provides support and customization for open source software of various kinds, primarily the Linux operating system.

We believe that regardless of the chosen licensing mechanism, open source software in the elections space must have two characteristics. First, jurisdictions who purchase software from a vendor should have full access to, and ownership of, the source code and be able to freely modify the software to serve their own future needs and to hire other companies or individuals to carry out such modifications on their behalf. Second, voters and other interested citizens should have full, immediate, and unimpeded access to inspect the source code. They should also be free to publicly state their observations about the source code and related artifacts, with no restrictions.

[1] `http://www.github.com/`

[2] Red Hat Inc. is a publicly traded company whose entire business model is based upon sales and support of Open Source software, primarily the Linux operating system. Red Hat's current market capitalization is just over $14 billion. `http://www.marketwatch.com/investing/stock/rht`

3 Open APIs and Data Formats

Most closed-source elections systems currently in existence are monolithic. In a monolithic system, the different parts cannot be distinguished or separated. This leads to various issues after systems have been purchased and put into service.

For example, imagine that a jurisdiction that has purchased a monolithic closed system likes their current ballot design system, but has decided based on audit results from multiple elections that the voting machine software is insecure. They have only two options: first, they can replace the entire system, losing the ballot design system they have already paid for; second, they can ask the vendor (that has already supplied them with software they dislike) to fix the system. In an system with open APIs and open data formats, there is a third option: the jurisdiction can reach out to a software vendor of their choice and request a drop-in replacement for the part of the system that they dislike.

Open APIs and data formats ensure that such a replacement can be developed by multiple providers. This not only gives the jurisdiction improved control over their final system, but also keeps the software vendors honest, stopping them from bundling subpar systems with more desirable systems simply because the two are inseparable.

Mandatory use of open APIs and data formats also opens the door for piecewise RFPs, where the state has the ability to choose different vendors to implement different parts of the system. This practice is now impacting the quality and nature of federal and state RFPs, as witnessed in the Healthcare.gov reboot at the federal level and in new compositional, agile, open interface-based RFPs coming out of the state of California.[3] Moreover, the U.S. Election Assistance Commission (EAC) and the National Institute of Standards and Technology (NIST) are moving toward a component-based certification process in the next version of the Voluntary Voting System Guidelines (VVSG).[4]

Consequently, we urge the State of Washington to expect and insist upon open APIs and data formats for all elections technology vendors.

[3] `http://www.codeforamerica.org/blog/2015/11/30/a-new-approach-to-procuring-government-technology-in-california/`

[4] `http://www.nist.gov/itl/vote/`

4 Commercial off-the-shelf (COTS) hardware

Elections software that only operates on proprietary hardware makes it impossible for a jurisdiction to change vendors without replacing their entire voting system and puts the jurisdiction at the mercy of a single vendor with respect to upgrade and maintenance pricing. However, running the software on COTS hardware eliminates this lock-in. Software upgrade costs are decoupled from hardware upgrade and maintenance costs, and both software (when using open APIs and data formats, as discussed above) and hardware can be provided by many possible vendors.

Pricing is also typically better for COTS hardware because of the extant competitive market for computers and tablets. We believe that the decision of which voting system to buy should be driven by the quality of the software. Why should a company that sells election software, which should be able to run on any reasonable hardware, be able to lock you into also purchasing their hardware at artificially inflated prices?

The decision to use COTS hardware or use a vendor that provides elections software-as-a-service is not without its challenges. The key challenge—especially in the context of mission-critical systems that mandate certification or rigorous quality assurance—is the means by which vendors prove to their clients and the public that the systems they deploy in an election are, in fact, the systems certified for use by the authorities. Solutions that do not rely upon trusted platform modules or rigorously engineered open source roots of trust should not be acceptable. Current VVSG requirements, which focus on publishing and self-reporting on hashes and similar mechanisms, are easily circumvented or falsified and are therefore not useful.

5 Evidence of correctness for unsupervised voting

It is critical for an elections system to ensure that those who do not vote at polling places, and thus are voting in an unsupervised fashion, still have evidence that their vote has been counted. This goal applies to electronic ballot delivery, electronic ballot marking tools, and vote by mail, early voting or UOCAVA.

Galois has already demonstrated the risks of allowing voters to submit votes electronically.[5] Many additional attacks on electronic ballot submission are possible, such as surreptitiously delivering a ballot to the voter that looks correct but in reality will be unreadable or deemed invalid by the voting system.

The existence of these classes of attack suggests that any unsupervised election system must be considered part of the core verifiable voting system. These systems must have extensive and rigorous security measures and be verifiable by the voter. Techniques exist that enable voters to verify that their vote has been counted without sacrificing privacy or requiring much extra effort or cost, either on the part of the election authorities or voters.[6] As vote by mail becomes increasingly popular around the country, and because these techniques give a quantum leap in assurance via voter verifiability, it should be expected that they will be a standard part of the best elections systems moving forward. Washington State should see this RFP as an opportunity to be a thought-leader to the world in this regard.

6 Audits

Even with any or all of the above techniques in place, each election as a whole, and all of its critical software components, should still be audited by a third party. That is, someone with no stake in the results of the election or the success of the system running the election should be given full access to examine its workings and results to make sure that the outcome announced by the authorities is beyond dispute.

These audits range from watching election setup procedures to ensure that they follow local statutes to statistically verifying the entire result of the election by examining only a relatively small number of randomly selected ballots. Any amount of auditing greatly enhances public confidence in elections and catches problems that might occur quickly enough that they are rectified before they cause an expensive or embarrassing recount.

Galois and its elections spin-out offer a full suite of auditing services using state of the art techniques. We welcome a further conversation about how we can be helpful to

[5] Details of this demonstration are available at `https://galois.com/blog/2014/11/hacking-internet-voting-via-ballot-tampering/`.

[6] One such technique is described in *Verifiable Postal Voting*, `https://doi.org/10.1007/978-3-642-41717-7_8`, by Josh Benaloh, Peter Y. A. Ryan, and Vanessa Teague.

Washington in providing additional confidence that its elections are held to the highest standards.

7 Summary

Everything we have described in this response can be implemented by a company that is familiar with building secure, correct systems from the ground up. Techniques for building systems in this manner have existed in mission-critical areas such as avionics, transportation, and national defense for decades. Our local and national elections should be treated with the same respect and care as other systems upon which our lives and livelihoods depend. In order to do this, we recommend that the eventual RFP not place inappropriate preconditions on potential applicants, and that it emphasize, or give favor to, solutions that have the following key aspects:

- open source software;
- open APIs and data formats;
- commercial off-the-shelf (COTS) hardware; and
- evidence of correctness for unsupervised voting

We believe that all of these aspects will increase public trust in voting systems. We also expect that systems exhibiting the first three will be higher quality and more affordable in the long term.